Amendments to the Specification

Please amend the paragraph bridging pages 3 and 4, in the following manner:

Fig. 1 is a drawing illustrating an example of a quantized 8 times 8 block register in a DCT coefficient algorithm. Labels r00, r01, r02, ... r76, r77 are applied in Fig. 1 to reference elements of the 8 times 8 block register in Fig. 1. Such labels are not shown in, but also apply to corresponding elements in, the 8 times 8 block registers of Figs. 5-7.

Zigzag arrows show a sequence of zigzag scanning at the time of Huffman coding. [[A]] In the example of Fig. 1, pixel r00 represents a directing, or DC component which shows intensities (i.e., brightness) of 8 times 8 blocks. The other components are AC components represented by a slope from a low-frequency component to a high-frequency component as indicated by a translucent arrow.

Please add the following paragraph between lines 6 and 7 of page 16:

Fig. 5A is a drawing illustrating the 8 times 8 block register of Fig. 5 after the coefficient 1 of the r13 pixel has been changed to 0, and the r22 pixel has been modified to a coefficient 1;.

Please amend the paragraphs at page 17, line 21 through page 18, line 5 in the following manner:

Fig. 4 is a flowchart illustrating an information compression method configured to obtain a decreased amount of coding [[of]] in the DCT coefficient encoder block 20 in Fig. 2. Fig. 5 is a drawing illustrating an inverse zigzag scan operation, in the method of Fig. 4, on a ~~of the~~ quantized 8 times 8 pixel block register in Fig. 4.

As indicated ~~by arrows of~~ in Fig. [[1]] 4, a correction level is set in the correction level setup register 22 (Step 101). The correction level refers to [[an]] a

coefficient for indicating a degree of data correction for setting the number of times of data correction or a value [[of]] or amount of coding. An increase in the correction level degrades image, but reduces the amount of coding to be generated.

Please amend the paragraphs at page 18, line 9 through page 19, line 1, in the following manner:

Referring now to Fig. 5, the 8 times 8 pixel block register 21 is inversely zigzag scanned to search for valid coefficients. In a conventional JPEG ~~algorism~~ algorithm, Huffman coding is performed in accordance with a zigzag scan order (i.e., in a forward direction) shown in Fig. 1. As a result, as one exemplary valid coefficient r13 shown in Fig. 5 (see r13 of Fig. 1), a code length of Huffman coding is increased when a front and back blocks of the coefficient r13 are invalid (coefficients with 0 value).

The first valid coefficient (r13 with 1 value) ~~initially searched by~~ found in the search in the inverse zigzag scan shown in the example of Fig. 5 is modified to 0 (Steps 103 and 104). At the same time, the correction counter 23 is counted up from 0 to 1 (Step 104).

When the correction counter 23 lacks a correction level or the number of ~~a coefficient~~ coefficients to be modified is greater than one or ~~the coefficient~~ such number exceeds a predetermined value even if one coefficient is modified from 1 to 0 (Step 105), the inverse zigzag scan continues and the next valid coefficient (r30 with 1 value) is modified to 0 (Steps 103 and 104).

Please amend the paragraphs at page 19, line 16 through page 20, line 5 in the following manner:

The above-mentioned flow may be applied to [[mula]] luminance or chroma components.

In Step 103, when a newly ~~searched~~ found valid coefficient 1 is located at a relatively low frequency or a coefficient itself has a large value, correction affects the image. To avoid these ~~affects~~ effects, a register which sets up minimum frequency or maximum valid coefficient for correction may be applied.

When the image deterioration exceeds a limit even when the above-mentioned techniques is applied, an address of an isolated valid coefficient may be moved to line up other valid coefficients together without modifying the valid coefficient to the invalid coefficient. [[For]] In the example [[, in]] of Fig. [[4]] 5, the r13 pixel has a coefficient 1 ~~. The coefficient 1~~ which is modified to 0. At the same time In another example (Fig. 5A), the coefficient 1 can be moved from r13 to the r22 pixel located in front of r13 ~~is modified to a coefficient 1~~ in the inverse zigzag scan.

Please amend the paragraphs at page 20, line 15 through page 22, line 8, in the following manner:

Fig. 6 is a drawing illustrating an example of ~~an separated~~ contents of the 8 times 8 pixel block register 21 according to another embodiment of the present specification. Fig. 8 is a flowchart for explaining an operation in the embodiment of Fig. 6.

In the above-mentioned embodiment shown in Fig. 2, when the 8 times 8 pixel block register 21 is inversely zigzag scanned, each register block is ~~required to be~~ accessed so that ~~substantially~~ a substantial amount of time is consumed. [[This]] However, the embodiment improves this point for efficiency of the scan time.

[[A]] The block register [[shown]] in the embodiment of Fig. 6 is provided with an OR circuit (i.e., block register nets f1 to f15) at the same time the coefficient is set to each block per scan lines or in accordance with each frequency (which is used by the DCT conversion) of a table after quantization.

The block register nets include OR circuits of all registers for each frequency (f1 to f15). [[Left]] The numbers to the left of the labels for f1 to f8 and the numbers below the labels for f9 to f15 represent results of OR circuits. Therefore, when one

pixel or one bit is input, the value of the block register net is 1. In this process, 1 is set to the block register nets f1 to f15 immediately once data are input in all 64 blocks so that a correction address is immediately selected without the inverse zigzag scan.

In this embodiment, the structure of an information compression apparatus is the same as Fig. [[1]] 2 except that the block ~~not registers~~ register nets f1 to f15 are added to the 8 times 8 pixel block register 21.

Referring to Fig. 8, [[an]] general operation of the information compression apparatus is illustrated.

The correction level is set in the correction level setup register 22 (Step 201). The quantization execution module 10 then performs a DCT transformation and latches the quantized data into the 8 times 8 block register 21 (Step 202). [[The]] A search for valid coefficient ~~is searched from~~ in the block register ~~blocks~~ nets included in the searched block register [[net]] is performed (Step 203). The block register is checked to search for the block ~~which closes~~ register net having a valid coefficient which is closest to [[the]] high-frequency. In the example of Fig. 6, the block register net f5 is ~~searched~~ such a net (Step 204). The initially searched valid coefficient (r13 with 1 value) is corrected to the invalid coefficient (r13 with 0 value). At the same time, the correction counter 13 is counted up from 0 to 1 (Step 205).

When the correction counter 23 lacks a correction level (Step 206), the search device 24 continues to inversely search the next block register net (Step 204) and the data correction device 25 modifies the next valid coefficient ([[r14]] r03 in f4 with 1 value) to 0 (Step 205).

Please amend the paragraphs at page 24, line 21 through page 25, line 12, in the following manner:

When the embodiments shown in Figs. 6 and 7 are applied to Fig. 3, the operation is the same steps as Fig. 5. In ~~order word~~ other words, the quantization execution module 10 performs the DCT transformation and outputs the quantized data into the 8 times 8 block register 21. At the same time, the search device 40

receives the data ~~to transfer~~ and transfers it to the search control device 24. As a result, the search control device 24 is prepared to operate the block register nets f1 to f15. In addition, when the embodiment shown in Fig. 7 is applied to Fig. 3, only block register nets f1 to f8 need be configured to operate for further reducing process time.

The block register 21 may include a calculating step for calculating a total sum of coefficients of block registers arranged along each scanning line corresponding to one of different frequencies used in the DCT frequency conversion algorithm (Figs. 6A), a summing up step for summing up a plurality of the total sums (Figs. 6A), and a start address changing step for changing an address of the block register to start the inverse zigzag scan.